

## **REMARKS**

Claims 1-3 and 5-34 are pending. Claims 1, 7, 17, 21 and 31 are independent. Claims 1-3 and 5-34 are rejected under 35 U.S.C. § 103. Reconsideration of the rejection in view of the following remarks is respectfully requested.

While no amendment was necessary to overcome the rejection, claims 1, 7, 17, 21 and 31 are commonly amended to add additional patentable distinctions over the cited references in order to proceed to allowance as quickly as possible. Support for the amendment may be found, for example, in paragraph 0061 of the published version of the application, i.e., Patent Application Publication No. 2005/0096927.

### **Telephone Conversation With Examiner Anya**

Examiner Anya is thanked for the telephone conversation conducted on November 16, 2009. Differences between claimed subject matter and the asserted art were discussed. Clarifying amendments were discussed to better place the application in condition for allowance. It appears that the rejections based on the asserted references are overcome.

### **Rejection of Claims 1-3 and 5-34 under 35 U.S.C. § 103(a)**

Claims 1-3 and 5-34 are rejected under 35 U.S.C. § 103(a) as being unpatentable over (1) U.S. Patent Application Publication No. 2003/0093500, by Khodabakchian *et al.* (hereinafter referred to as “Khodabakchian”) in view of at least one of: (2) U.S. Patent No. 7,036,045, issued to Broussard (hereinafter referred to as “Broussard”); and (3) U.S. Patent Application Publication No. 2003/0204835, by Budhiraja *et al.* (hereinafter referred to as “Budhiraja”). Specifically, claims 1, 2, 5-10, 12-24 and 26-34 are rejected in view of Khodabakchian and Broussard and claims 3, 11 and 25 are rejected in view of Khodabakchian, Broussard and Budhiraja. (Office Action, pp. 2-9.) Applicants respectfully traverse the rejections.

It is respectfully submitted that the combination of references, even if proper, fails to teach or suggest the claimed subject matter. In summary, the Office Action admits

Khodabakhchian fails to teach or suggest the claimed subject matter because it teaches a centralized exception handler 202 (FIG. 2), but alleges that Broussard teaches an instance having exception handling code or failure handling functionality. Applicants point out in detail below that Broussard, like Khodabakhchian, fails to teach an instance having exception handling code or failure handling functionality.

Khodabakhchian discusses a system “handling processes that interact with one or more asynchronous systems.” Khodabakhchian at ¶ 0005. Khodabakhchian states that, “Web service orchestration server 102 contains multiple scenarios 110(1), 110(2) and 110(3).” Khodabakhchian at ¶ 0023. “A scenario is a programming abstraction of a long-running process or a collaborative process.” Id. “An exception handler 202 processes exceptions that are generated by orchestration engine 200 when implementing the instructions and commands in a scenario 110. Exceptions may include, for example, a fault generated by a web service or a notice generated as a result of a web service timeout.” Khodabakhchian at ¶ 0025; Figure 2. Khodabakhchian indicates that failures are not processed within an instance of a process, but instead are processed by an independent “exception handler” 202 on web service orchestration server 102. Id.

The Office Action admits Khodabakhchian fails to teach or suggest the claimed subject matter. (Office Action, p. 4). Contrary to the admission in the Office Action, Khodabakhchian is not merely “silent” on the subject of exception handling code in an instance. Khodabakhchian clearly indicates that exception handling in a separate centralized process 202 (FIG. 2) that is not present in scenarios.

The Office Action alleges that Broussard teaches what Khodabakhchian fails to teach. However, it is respectfully submitted, Broussard fails to teach what the Office Action says it does. Specifically, Broussard states as follows:

Problems often occur, however, because many applications write code that catch exceptions that perform some type of default behavior. In Java, for example, exceptions are sometimes caught generically, such as with the statement:

catch (Exception e) {/\* do nothing \*/}.

In this case, nothing is done except to consume the exception. While this may work fine for cases where the exceptions are intended, it may be insufficient to handle other, perhaps unintended, exceptions such as a ClassNotFoundException. In cases where the exception is due to some user error (e.g., configuration problem, missing classpath item, etc.), the program will behave badly, and there may be no visible sign of what the problem is.

Broussard, col. 1, ll. 30-43 (emphasis added).

Within application logic 50 is a piece of code containing a "try-catch" block 56 to deal with any "Exception e" errors that might occur while "Do some business logic" is executed. If an error occurs 58, an exception is created 60, and the verbose exceptions mode is checked 62. If verbose is enabled for the exception 64, then the exception is output along with its stack trace 66, and the application logic continues 52.

Broussard, col. 4, ll. 57-64.

```
Try {  
  Do some business logic  
} catch (Exception e)  
  {do nothing}
```

FIG. 4 (block 56).

As stated in Broussard's background section, "do nothing" means "nothing is done except to consume the exception." Thus, nothing is done in the application logic, 50, 52, let alone JVM 18. The try-catch block 56 is not even in the application logic. The try-catch block 56 is part of JVM 18 with exception dumping system 24. Input file "App" 31 is separate and is shown as application logic 50, 52 in FIG. 4. FIG. 4 clearly shows that catching the exception and the dumping system has nothing to do with the application logic 50, 52. Furthermore, there is no error handling. Instead, steps 58-66 are only error logging, not error handling. There is no error compensation. FIG. 4 shows that error occurs 58 then exception created 60 then check mode 62 then if verbose enabled for exception 64 output exception/stack trace 66. Broussard makes quite clear that nothing is done upon catching the exception, i.e., "do nothing." Like

Khodabakchian, any handling code specifying error compensation is elsewhere, not in the application logic 50.

Additionally, there is no legitimate motivation to modify Khodabakchian. Khodabakchian, like other references, handles errors in a separate process using centralized exception handler 202. The Office Action asserts that the motivation to change this so that Khodabakchian's scenarios handle exceptions is to "provid[e] a condition system with a mechanism for signaling and handling unusual conditions, including errors and warnings." This generic conclusion is clearly without merit because Khodabakchian already handles unusual conditions separately using Exception Handler 202.

The foregoing remarks apply equally well to all pending claims. Further, the Office Action admits that Khodabakchian fails to disclose storing the instance after a predetermined time as claimed in claims 3, 11 and 25, but alleges Budhiraja does. (Office Action, p. 9). It is respectfully submitted that this argument is also inaccurate.

The Office Action alleges that paragraphs 0037, 0041 and 0048 of Budhiraja teach, as claimed in claims 3, 11 and 25, "wherein storing the instance takes place after a predetermined time." Specifically, the Office Action alleges use of the word "checkpoint" teaches claims 3, 11 and 25. However, nowhere does Budhiraja teach what is claimed. Checkpointing discloses nothing about setting a predetermined time to store an instance. Here is what Budhiraja actually says:

[0037] Business process manager 408 has the ability to persistently save (checkpoint) and recover the state of the business process objects it creates. For the particular implementation shown, business process manager 408 uses a database (data store) for this purpose. Other implementations may use different persistent storage methods.

[0041] The checkpointing ability of Business process manager 408 is extended so that name and label information is included with saved state information. This means that each time that business process manager 408 saves the state of a business process object, it also saves the name and label associated with the business process model used to create that business process object.

[0048] Process modeler 402 and BusinessWare server 404 are configured so that sub-process models may be labeled in the same way as business process models. Thus, there may be multiple versions of a sub-process model. Each definition may have an associated label including an "initial" and a "current" version. The checkpointing ability of Business process manager 408 is configured to include the version information for sub-process models during state saving operations. This allows business process objects to be restarted using the correct versions of their nested sub-process models.

Thus, Budhiraja does not provide the disclosure missing in Khodabakhchian and Broussard as to claims 3, 11 and 25. Further still, Budhiraja does not store instance states, remove instances from memory and then restore them. Budhiraja merely stores states while maintaining business process objects in active memory. These are additional reasons the rejection should be withdrawn against claims 3, 11 and 25.

It is respectfully submitted that the claimed subject matter is allowable over the cited references. Accordingly, Applicants respectfully request reconsideration and withdrawal of the rejection of claims 1-3 and 5-34 under 35 U.S.C. § 103(a) in view of various combinations of Khodabakhchian, Broussard and Budhiraja.

**DOCKET NO.:** MSFT-2748/302029.01  
**Application No.:** 10/698,762  
**Office Action Dated:** September 23, 2009

**PATENT**

### **CONCLUSION**

Amendments, made herein or previously made, are without abandonment of subject matter. Applicant expressly reserves the right to, in the pending application or any application related thereto, reintroduce any subject matter removed from the scope of claims by any amendment and introduce any subject matter not present in current or previous claims.

In view of the foregoing remarks, it is respectfully submitted that this application is in condition for allowance. Reconsideration of this application and an early Notice of Allowance are respectfully requested.

Date: December 22, 2009

**/Joseph F. Oriti/**  
Joseph F. Oriti  
Registration No. 47,835

Woodcock Washburn LLP  
Cira Centre  
2929 Arch Street, 12th Floor  
Philadelphia, PA 19104-2891  
Telephone: (215) 568-3100  
Facsimile: (215) 568-3439